RESEARCH PAPER
CAA-RP-87-4

2

AN ALGORITHM FOR CALCULATING THE AREA OF OVERLAP OF AN ELLIPSE AND A CONVEX POLYGON

NOVEMBER 1987

CONCEPTS ANALYSIS AGENCY
UNITED STATES ARMY

PREPARED BY
DR. ROBERT L. HELMBOLD

US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20814-2797

CAA

DISCLAIMER

The findings of this report are not to be construed as an official Department of the Army position, policy, or decision unless so designated by other official documentation. Comments or suggestions should be addressed to:

Director
US Army Concepts Analysis Agency
ATTN: CSCA-MV
8120 Woodmont Avenue
Bethesda, MD 20814-2797

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) CAA-RP-87-4	5. MONITORING ORGANIZATION REPORT NUMBER (S) N/A
6a. NAME OF PERFORMING ORGANIZATION US Army Concepts Analysis Agency 6b. OFFICE SYMBOL (if applicable) CSCA-MVM	7a NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) 8120 Woodmont Avenue Bethesda, MD 20814-2797	7b ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Concepts Analysis Agency 8b OFFICE SYMBOL (if applicable) CSCA-MVM	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) 8120 Woodmont Avenue Bethesda, MD 20314-2797	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO. / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
An Algorithm for Calculating the Area of Overlap of an Ellipse and a Convex Polygon

12 PERSONAL AUTHOR(S)
Helmbold, Dr. Robert L.

13a TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Research Paper	FROM 10/87 TO 11/87	1987 November	

16. SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Military Operations Research, Nuclear Weapons Simulations, Wargaming, Combat Simulation.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper describes a new and improved algorithm for estimating in computer simulations the area of overlap of an ellipse and a convex polygon. The need for such algorithms arises frequently in military operations analysis, in particular in estimating the portion of a rectangular target overlapped by a disk-shaped nuclear coverage area. A detailed description of the algorithm and illustrations of its application are included.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Helmbold	22b TELEPHONE (Include Area Code) (202) 295-5226 22c OFFICE SYMBOL CSCA-MVM

DD Form 1473, JUN 86 Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

RESEARCH PAPER
CAA-RP-87-4

AN ALGORITHM FOR CALCULATING THE AREA OF OVERLAP OF AN ELLIPSE AND A CONVEX POLYGON

November 1987

A-1

Prepared by

Dr. Robert L. Helmbold

US Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, Maryland 20814-2797

PREFACE

This research paper presents an efficient new algorithm for computing the exact area of overlap of an ellipse and a convex polygon. Its fundamental idea originated with Dr. Dennis F. DeRiggi of the US Army Concepts Analysis Agency (CAA), who used it to develop an algorithm for computing the exact area of overlap of a disk and a rectangle. Dr. Robert L. Heimbold, also of CAA, subsequently observed that DeRiggi's approach generalized in an obvious way to ellipses and convex polygons.

The problem of finding the area of overlap of two generic figures of prescribed shapes, sizes, orientations, and locations arises frequently in military operations research. Damage functions are often modeled as circular "cookie-cutters," and targets are frequently represented as disks or rectangles. Convenient formulas or algorithms are available for calculating exactly the area of overlap of two disks, or of two rectangles oriented so that their sides are parallel. They appear to be widely known and are often used by the military OR community.

In addition, several methods for numerically approximating the area of overlap of a disk and a rectangle are available and widely used. However, they can be wildly inaccurate for certain configurations of the disk and rectangle, none generalize easily to the case of an ellipse and convex polygon, and they can be impractically slow and expensive when very high accuracy is required. Hence, an efficient algorithm for calculating the exact area of overlap of an ellipse and a convex polygon would be useful in many military OR simulations and analyses.

Such an algorithm is described in this research paper.

CAA-RP-87-4

DEPARTMENT OF THE ARMY
US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20814-2797

REPLY TO
ATTENTION OF

CSCA-MVM

14 SEP 1988

MEMORANDUM FOR: Deputy Under Secretary of the Army (Operations Research), The Pentagon, Room 2E660, Washington, D.C. 20310

SUBJECT: An Algorithm for Calculating the Area of Overlap of an Ellipse and a Convex Polygon

1. This Research Paper grew out of an earlier effort begun by Dr. Helmbold to find an improved algorithm for approximating the area of overlap of a disk and a rectangle. This paper stimulated Dr. DeRiggi to devise a greatly improved and very ingenious simple exact method for computing those kinds of overlap areas. When this method was communicated to Dr. Helmbold, he quickly saw that easy generalizations of Dr. DeRiggi's basic idea would yield an excellent algorithm for calculating the area of overlap of an ellipse and a polygon.

2. This algorithm will help to increase the accuracy and efficiency of DOD military operations analyses, for they frequently encounter the problem of calculating such overlap areas. For example, ellipses are often used to represent the area of effect of a weapon and a polygon is used to approximate the target's configuration. Questions or inquiries should be directed to our Office of Special Assistant for Model Validation, U.S. Army Concepts Analysis Agency (CSCA-MVM), 8120 Woodmont Avenue, Bethesda, MD 20814-2797, (301) 295-1669.

Encl

E. B. VANDIVER III
Director

CAA	AN ALGORITHM FOR CALCULATING THE AREA OF OVERLAP OF AN ELLIPSE AND A CONVEX POLYGON	STUDY SUMMARY CAA-RP-87-4

THE REASON FOR PERFORMING THIS STUDY was to develop and document an improved algorithm for determining in computer simulations the area of overlap of an ellipse and a convex polygon.

THE PRINCIPAL FINDINGS are that a useful algorithm can be developed for determining in computer simulations the area of overlap of an ellipse and a convex polygon. It appears to be a new method offering many advantages over those previously proposed.

THE MAIN ASSUMPTION is that the polygon is convex.

THE PRINCIPAL LIMITATION is that numerical roundoff error may, under some conditions, reduce the accuracy of the result.

THE SCOPE OF THE WORK is limited to determining the area of overlap of an ellipse and a convex polygon.

THE RESEARCH OBJECTIVE was to develop and document an algorithm for determining in computer simulations the area of overlap of an ellipse and a convex polygon.

THE WORK WAS SUPPORTED BY the US Army Concepts Analysis Agency.

Tear-out copies of this synopsis are at back cover.

CONTENTS

FIGURES

TABLES

CHAPTER 1

EXECUTIVE SUMMARY

1-1. **PROBLEM.** Develop an algorithm which, given an ellipse and a (convex) polygon, calculates their area of overlap, i.e., the area common to both the ellipse and the rectangle. For example, in Figure 1-1 the area of overlap is shown as a shaded region.

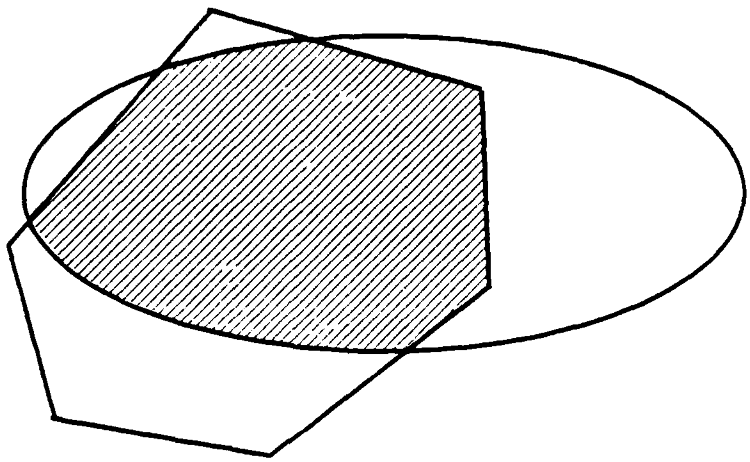

Figure 1-1. Area of Overlap

1-2. BACKGROUND

a. **Need.** The need for estimating the area of overlap of a disk and a rectangle, which is a special case of an ellipse-polygon overlap, arises frequently in computer simulations of military operations. Rectangles are often used to represent regions occupied by troops, civilian population centers, or other target elements, while disks are used to represent damage zones of nuclear weapons. For instance, the Nuclear Fire Planning and Assessment Model (NUFAM) used at the US Army Concepts Analysis Agency (CAA) employs rectangles to represent military unit positions and disks to represent nuclear effects coverage zones. Since rectangles and disks are frequently used in this way, the problem of estimating their area of overlap is a familiar one and various solutions to it have been proposed. A few of these are mentioned below with no attempt to be exhaustive. Presumably, many others have been put forward at one time or another.

b. **Theoretically Exact Procedures.** It is, in principle, possible to decompose the area of overlap into a finite number of figures bounded by circular arcs and straight lines. Then mensuration formulas or integral calculus can be applied to find the area of each figure, and the area of overlap obtained as their sum. This method provides theoretically exact results, since no numerical approximations other than those involved in finite arithmetic precision are required. However, attempts to program this approach on computers have shown it to be extremely cumbersome. The practical difficulty is that a large number of special cases must be considered, depending on how the disk intercepts the rectangle. Hence the algorithm must be arranged to determine which of the special cases applies, to subdivide the area of overlap appropriately, and to apply the correct mensuration formula to each subdivision. This proves exceedingly tedious to implement. Moreover, this approach does not generalize readily to the overlaps of polygons with either disks or ellipses.

c. **Replace Rectangle with an Array of Points.** In this approach the rectangle is replaced by an array of regularly spaced points, each of which represents a smaller rectangle. The area of overlap is approximated by multiplying the fraction of points within the disk by the area of the original rectangle. This is illustrated in Figure 1-2, where the area of overlap is estimated as 3/9 = 1/3 of the area of the original rectangle. The method is easy to understand and simple to program. However, its accuracy is low when the disk is small compared to the rectangle, and when few points are used in each rectangle. Increasing the number of points used in each rectangle sharply increases the computational time required. Moreover, the generalization to ellipses and polygons would be cumbersome and tedious.

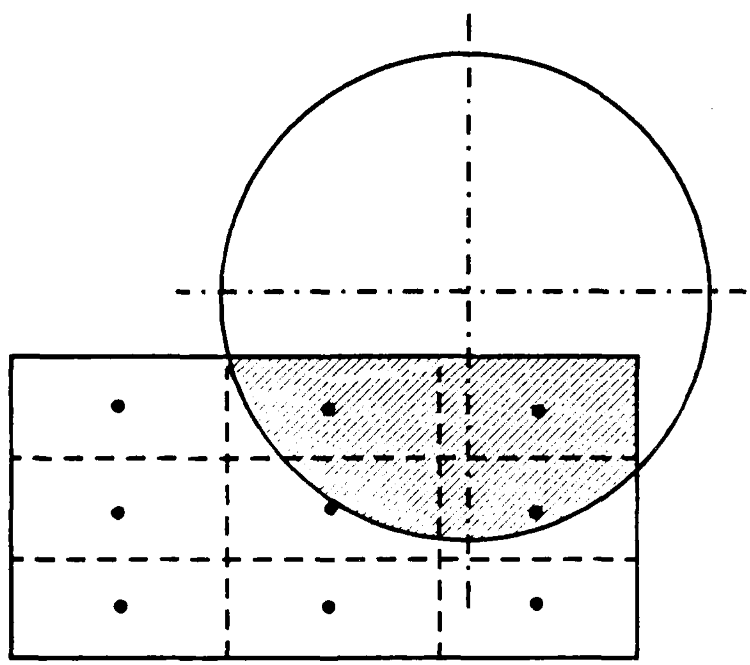

Figure 1-2. Replace Rectangle with a Regular Array of Points

d. Random Points in Rectangle. Here a certain number of points in the rectangle are selected randomly, and the area of overlap estimated as the fraction of them inside the disk. This Monte Carlo approximation to the area of overlap has all the usual advantages and disadvantages of that method. In particular, many points are needed if the result is to be numerically accurate and statistically reliable. Moreover, the generalization to ellipses and polygons is not evident.

e. Functional Approximation. This approach begins by considering the characteristic functions of the rectangle and the disk, where the characteristic function of any region is defined to be equal to one for points in the region, and zero elsewhere. Then the area of overlap of any two regions is just the integral of the product of their characteristic functions. Since the product of the characteristic functions vanishes outside the area of overlap, the integral may be carried out over all space, and that is at least a useful theoretical simplification. In the functional approximation method, the (discontinuous) characteristic functions of the rectangle and disk are approximated by continuous functions. The functions used can be polynomials, Fourier series, orthogonal series of various types, and so forth. One difficulty with this approach is that it often is hard to estimate the accuracy of the result. Another is that several terms must be carried in the functional approximation. Moreover, a familiarity with advanced mathematical methods is needed to understand the method and to implement it properly. Finally, the generalization to ellipses and polygons would add measurably to the complexity of this approach.

f. Curve Fitting. Here a number of trial cases are accurately computed, and the results are tabulated against such key parameters of the situation as the rectangular dimensions of the rectangle, the offset of the disk's center from the center of the rectangle, and radius of the disk. (Usually these are normalized to some standard dimension, such as the disk's radius.) A more or less arbitrary function is then fitted to the tabulated values, and used to estimate the area of overlap for other values of the key variables. This is simple in conception. However, it is not easy to determine whether the functional form used for the fitting process gives satisfactorily accurate values for situations different from those used to obtain the tabulated values. Moreover, it depends on having a number of cases for which the area of overlap is accurately known, and so to that extent assumes what is to be found. Finally, curves fitted to the disk-rectangle case have no application to the general ellipse-polygon case.

g. Replace Disk by an Approximating Family of Rectangles. Here the disk is replaced by a family of rectangles, as illustrated in Figure 1-3. Now, it turns out that there is a fast, finite, simple algorithm for calculating the area of overlap of two rectangles whose sides are parallel. This algorithm is applied to find the area of overlap between the original rectangle and each member of the approximating family of rectangles. The sum of these individual areas is an estimate of the area of overlap of the disk and the rectangle. This method is often surprisingly accurate with even a few (12 to 20) rectangles in the approximating family. However, it is not a theoretically exact method, and it does not generalize readily to the ellipse-polygon case.

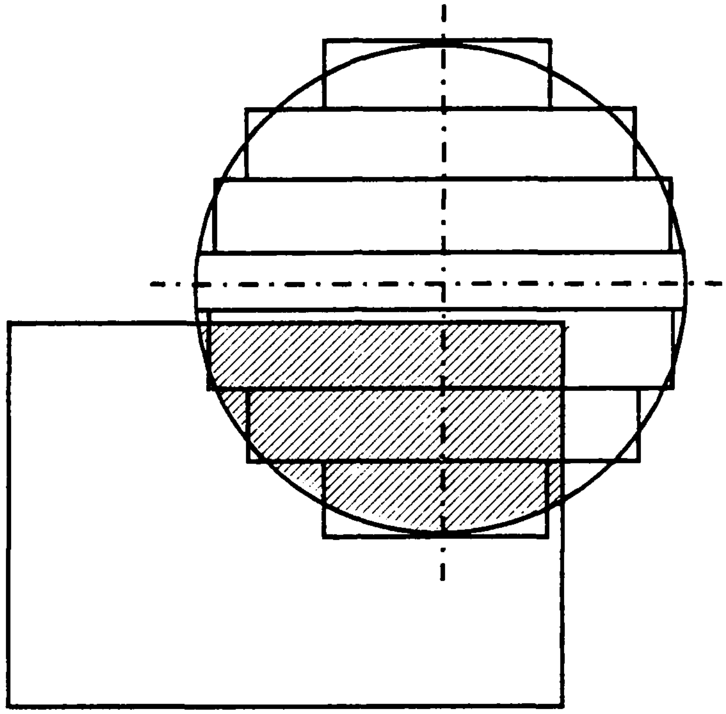

Figure 1-3. Replace Disk with a Family of Rectangles

1-3. SCOPE. This paper develops an algorithm for calculating the area of overlap of an ellipse and a convex polygon. This algorithm is believed new, and it differs from those mentioned in paragraph 1-2. It has several advantages over those previously proposed.

1-4. LIMITATIONS. The principal limitation is that numerical roundoff error may under some conditions reduce the accuracy of the result.

1-5. TIMEFRAME. Not applicable.

1-6. KEY ASSUMPTIONS. The key assumption is that the polygon is convex.

1-7. APPROACH

a. The approach used is to reduce, by a series of stages, the original problem to a finite number of similar problems, all of which are conceptually similar so that a single algorithm easily solves each of them. The key concepts involved at each stage of this reduction are very briefly summarized here, and are described more fully in Chapter 2.

b. First, any ellipse-polygon overlap case can be reduced to an equivalent disk-polygon overlap situation by an affine transformation of coordinates that shrinks the major axis of the ellipse until its equal to the minor axis. Such a transformation clearly maps the original convex polygon onto some other, but also convex, polygon. So the problem has now been reduced to one of finding the area of overlap of a disk with the interior of a convex polygon.

c. Second, observe that since the area of the disk is known, its area of overlap with the interior of the polygon could easily be found if we knew its area of overlap with the exterior. We now focus on finding the disk's area of overlap with the exterior of the polygon.

d. Third, to facilitate our work, the boundary of the new convex polygon is given a definite orientation. In this paper we will always use the conventional mathematical orientation according to which the counterclockwise direction is considered positive. Figure 1-4a shows a convex polygon with its vertices numbered in positive order. This orientation of the polygon orients, or gives a positive direction to, each of its sides.

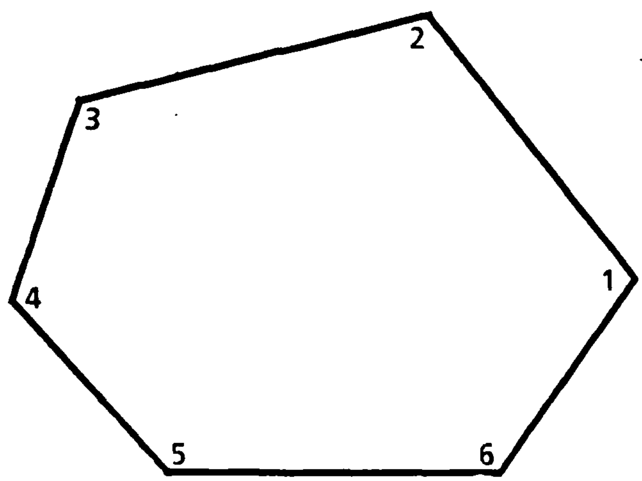

Figure 1-4a. Convex Polygon with Vertices Numbered in Positive Order

e. Fourth, to facilitate our work, each edge of the polygon is, by geometric construction, extended indefinitely in its positive direction. Figure 1-4b shows this construction. Observe that these construction lines partition the region exterior to the polygon into a finite number of unbounded but disjoint (i.e., nonoverlapping) wedge-shaped regions. (Here the polygon's convexity plays a crucial role. If the polygon is not convex then the construction of Figure 1-4b leads to wedge-shaped regions that are not disjoint.) There is exactly one such wedge-shaped region for each vertex of the polygon.

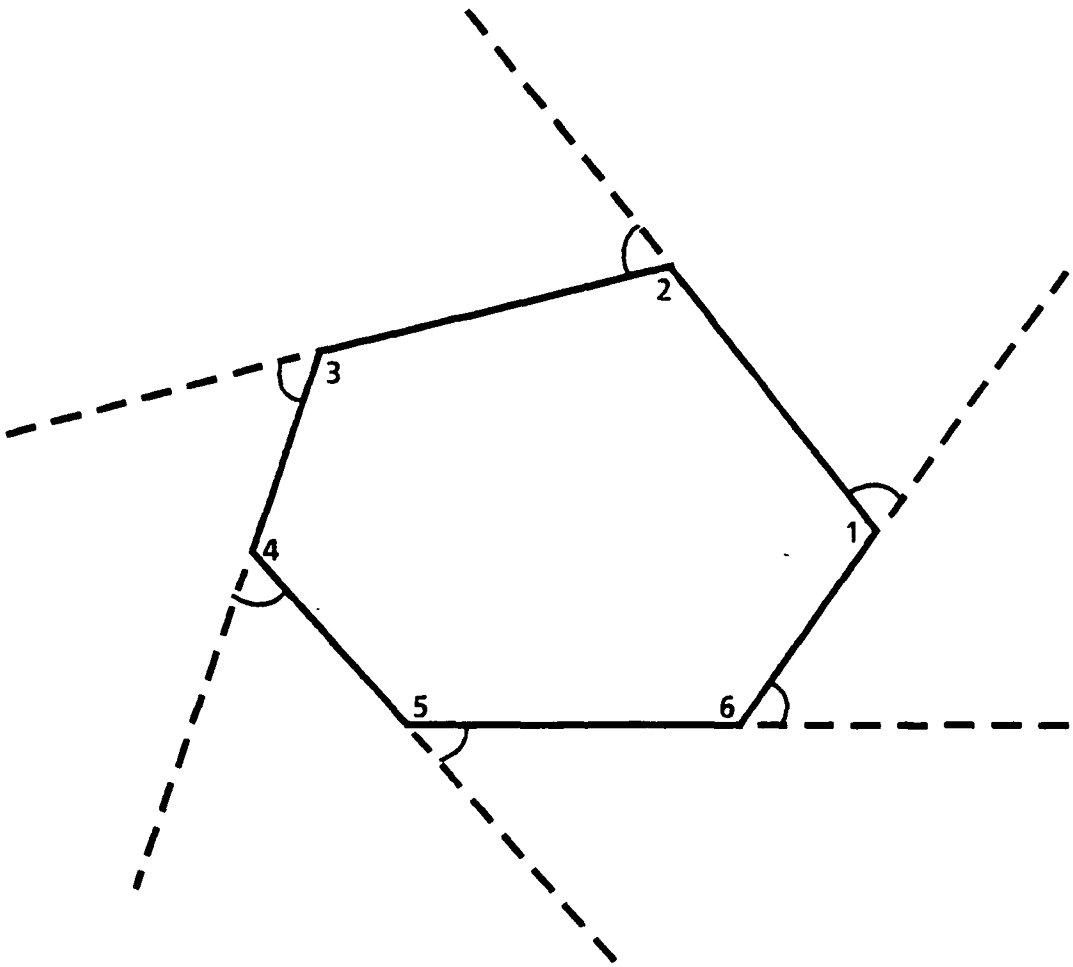

Figure 1-4b. Construction Lines

f. Fifth, observe that the construction of the last step reduces the problem (of finding the area of overlap of a disk with the exterior of a convex polygon) to one of finding the overlap of a disk with a typical wedge-shaped region. The area of overlap of the disk with the exterior of a convex polygon is simply the sum of its areas of overlap with each of the wedge-shaped regions constructed as shown in Figure 1-4b. (The validity of this statement depends crucially on the disjointness of the wedge-shaped regions, and hence on the convexity of the polygon.)

g. Sixth, it happens that there is a simple, convenient, finite, theoretically exact algorithm for calculating the area of overlap of a disk and a wedge-shaped region of the sort shown in Figure 1-4b. Hence, the problem of finding the individual disk-wedge overlap areas can be considered as easily solved. This completes our reduction of the original problem to a finite number of similar problems, all of which are conceptually similar, so that a single algorithm easily solves each of them.

1-8. CONCLUSIONS. A simple finite algorithm suffices to calculate the area of overlap of an ellipse and a polygon. The resultant algorithm is widely applicable in the sense that it gives theoretically exact answers for all possible configurations of the ellipse and the polygon. It is user-friendly in the sense that its implementation involves a straightforward computation whose net result is easily discernable. As a result, the algorithm should be easy to verify, debug, modify, or incorporate confidently into larger programs. The accuracy of the computation can be maintained by using double-precision arithmetic to control roundoff errors.

1-9. OBSERVATIONS

a. Practical implementation of the algorithm would be aided by developing subroutines--optimized for speed--for incorporation into large simulations or wargames such as NUFAM, CEM, FORCEM, COSAGE, VIC, and others.

b. The method yields the exact area of overlap of an ellipse with a convex polygon. Since a triangle is convex, the method is in principle capable of being applied to find the exact area of overlap of an ellipse with any finite region that can be triangulated, i.e., with any polygon. Triangulation is not actually necessary--any decomposition of the polygonal region into disjoint convex polygons is sufficient.

c. Many regions that arise in practice can be approximated satisfactorily by polygons. Hence, the method can in principle be applied to approximate the area of overlap of an ellipse with a fairly arbitrary region of the sort that often arises in practice.

d. Although it may not be the most efficient algorithm for the purpose, the method can in principle be used to find the area of any convex polygon. All that is necessary is to find the area of overlap of the polygon with a disk whose radius is sufficiently large that it completely covers the polygon. If the area of overlap does not change when the disk's radius is increased slightly, then the radius is sufficiently large.

CHAPTER 2

APPROACH

2-1. INTRODUCTION. We start with the ellipse and polygon in general configuration in a two-dimensional Cartesian coordinate system, as shown in Figure 2-1. This chapter explains how the general configuration is reduced to a disk-polygon overlap situation. The mathematical details of the further reduction to a finite number of disk-wedge overlap problems are provided in Appendix A. A computer program to implement the solution is presented in Appendix B. Examples of overlap areas computed using this computer program are given in Chapter 3.

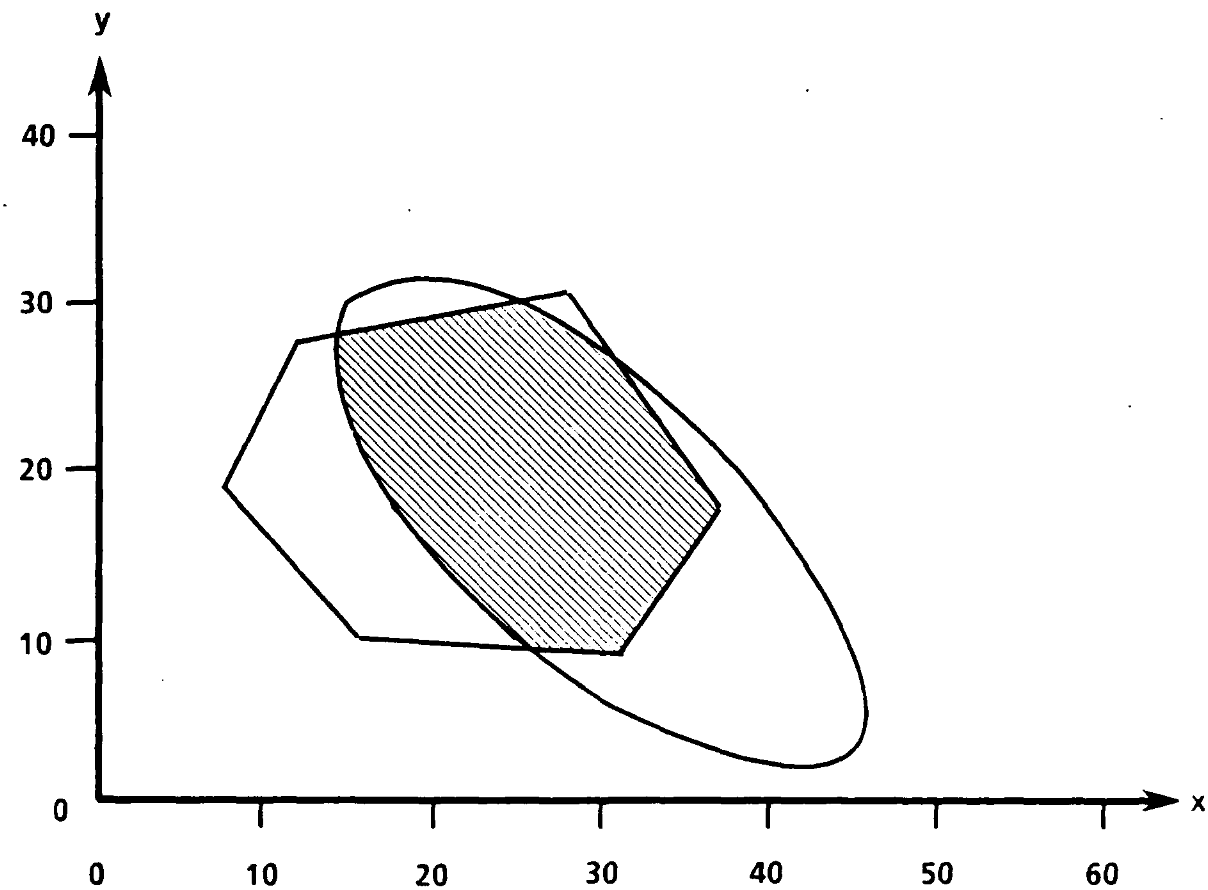

Figure 2-1. Ellipse and Polygon in General Configuration

2-2. DESCRIBE THE CONFIGURATIONS OF THE POLYGON AND ELLIPSE. The first step is to describe the configurations of the polygon and the ellipse relative to a two-dimensional Cartesian coordinate system. By referring to Figure 2-1, we see that this can be done by specifying the following items:

a. For the polygon, the number and coordinates of its vertices. (It is essential that the vertices be listed in positive order, as the algorithm assumes this.) Let N be the number of vertices in the polygon, and let (PX(J), PY(J)) for J = 1 to N be the Cartesian coordinates of the vertices, listed in positive order.

b. For the Ellipse

(1) The coordinates (XE, YE) of its center.

(2) Its semi-major and semi-minor axes (RX and RY, respectively).

(3) Its orientation angle, i.e., the angle (TDEG) from the x-axis to its major axis, in degrees. The algorithm assumes that TDEG is in the range $-90° < \text{TDEG} \leq +90°$.

2-3. REDUCE TO DISK-POLYGON OVERLAP SITUATION

a. Affine Transformation. The reduction is accomplished by an affine transformation that shifts the origin of the coordinate system to the center of the ellipse, rotates it so that the new x-axis is parallel to the major axis of the ellipse, and then shrinks the ellipse's major axis until it equals the minor axis. The algorithm to do that is described below.

b. Shift Origin and Rotate Axes. For each vertex, put

```
X(J) = + (PX(J) - XE) * Cos (TDEG) + (PY(J) - YE) * Sin (TDEG)
Y(J) = - (PX(J) - XE) * Sin (TDEG) + (PY(J) - YE) * Cos (TDEG)
```

c. Shrink New x-Axis. For each vertex, put

```
X(J) = X(J) * RY/RX
```

d. State Disk Radius. Put the disk radius, R = RY.

e. Results. Figure 2-2 shows the transformed (disk-polygon) configuration. Because of the shrinkage involved, the disk-polygon overlap area is (RY/RX) times that of the original ellipse-polygon configuration. To correct for this shrinkage, the algorithm at a later stage multiplies the disk-polygon overlap area by (RX/RY), the inverse of the shrinkage factor.

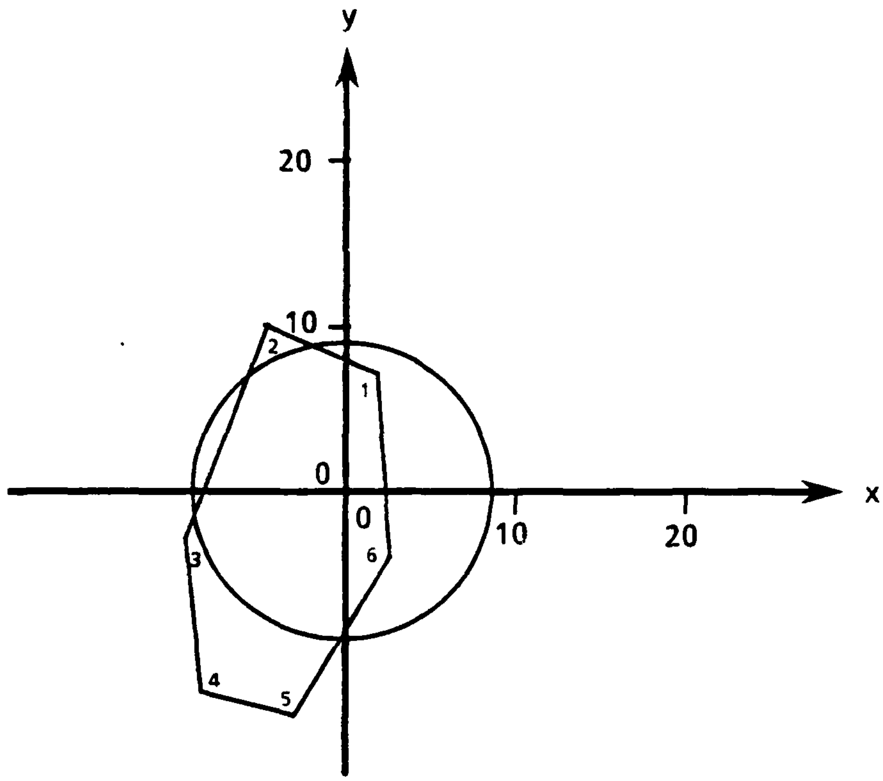

Figure 2-2. Disk-Polygon Overlap Configuration Obtained by Applying Affine Transformation to Figure 2-1

2-4. REDUCE DISK-POLYGON OVERLAP TO A SEQUENCE OF DISK-WEDGE OVERLAPS. Appendix A shows how to further reduce the disk-polygon overlap situation to a sequence of disk-wedge overlaps, and explains how to complete the solution.

CHAPTER 3

RESULTS

3-1. INTRODUCTION. This chapter presents some examples of overlap areas computed using the method expounded in this paper. In this chapter, all results are given to six significant digits.

3-2. EXAMPLE 1. Suppose that the configuration is that of Figure 2-1 and Table 3-1. Then the ellipse-polygon overlap area is equal to 321.392. The area of the polygon is 429.390, and the area of the ellipse is 565.487.

Table 3-1. Values for the Configuration for Example 1 (Figure 2-1)

Item	Symbol	Value
Ellipse center	XE	29.7
Ellipse center	YE	16.1
Ellipse semi-major axis	RX	20.0
Ellipse semi-minor axis	RY	9.0
Ellipse orientation angle	TDEG	-41.5°
Polygon Vertex No. 1	PX(1)	37.0
Polygon Vertex No. 1	PY (1)	17.9
Polygon Vertex No.2	PX(2)	28.5
Polygon Vertex No. 2	PY(2)	30.2
Polygon Vertex No 3	PX(3)	12.0
Polygon Vertex No. 3	PY(3)	27.2
Polygon Vertex No. 4	PX(4)	8.1
Polygon Vertex No.4	PY(4)	18.5
Polygon Vertex No. 5	PX(5)	15.5
Polygon Vertex No. 5	PY(5)	10.0
Polygon Vertex No. 6	PX(6)	31.0
Polygon Vertex No. 6	PY(6)	9.4

3-3. EXAMPLE 2. Disk-rectangle overlap areas may be the most common application of the method expounded in this paper. Accordingly, consider the configuration of Figure 3-1 and Table 3-2. The vertices of the rectangular polygon can easily be found from the information given. Alternatively, the configuration shown can be converted by a simple rigid coordinate translation and rotation to one in which the rectangle is centered at the origin and has sides parallel to the coordinate axes--a configuration in which it is easy to read off the coordinates of the vertices. In either case, the disk-rectangle overlap area for this example is 903.827.

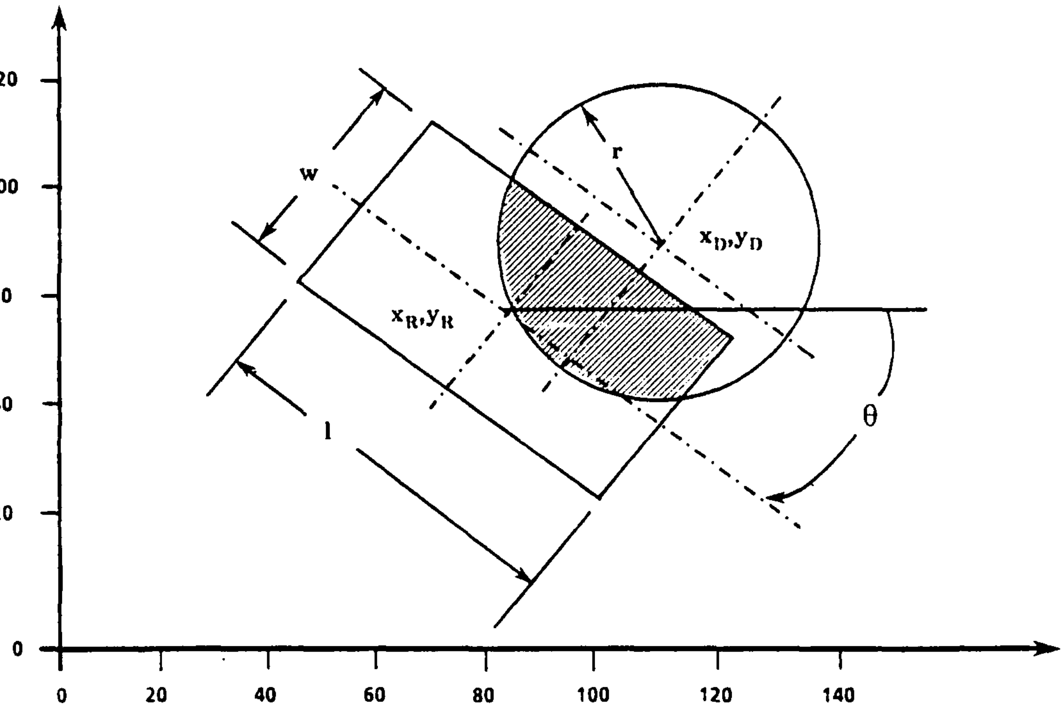

Figure 3-1. Configuration for Example 2

Table 3-2. Values for the Configuration of Example 2 (Figure 3-1)

Item	Symbol	Value
Rectangle center, x-coordinate	x_R	80.5
Rectangle center, y-coordinate	y_R	57.5
Rectangle angle	θ	-42°
Rectangle length	l	73
Rectangle width	w	37.5
Disk center, x-coordinate	x_D	111
Disk center, y-coordinate	y_D	66.5
Disk radius	r	31

3-4. **EXAMPLE 3.** For the configuration of Figure 3-2 and Table 3-3, the disk-rectangle overlap area is 131.769.

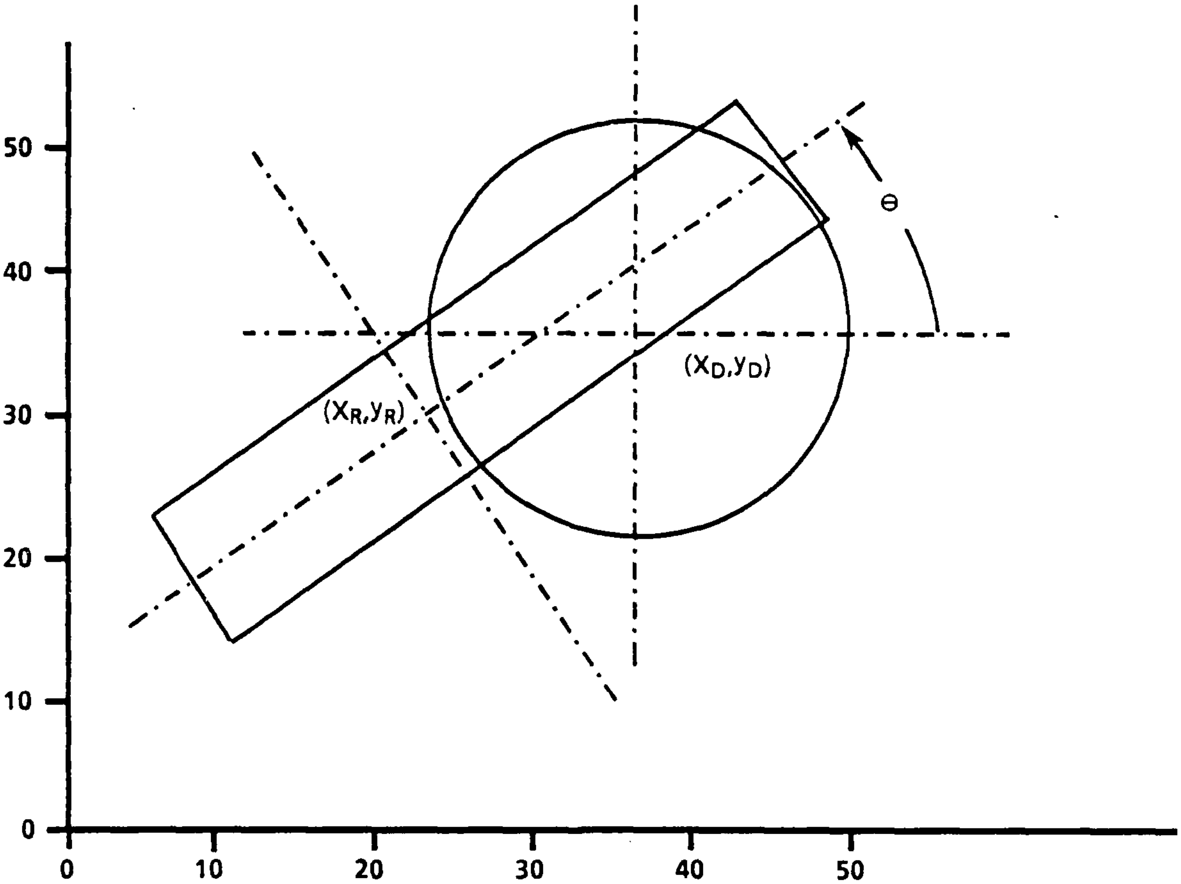

Figure 3-2. Configuration for Example 3

Table 3-3. Values for the Configuration of Example 3 (Figure 3-2)

Item	Symbol	Value
Rectangle center, x-coordinate	x_R	25
Rectangle center, y-coordinate	y_R	32
Rectangle angle	θ	33°
Rectangle length	l	44
Rectangle width	w	7
Disk center, x-coordinate	x_D	36
Disk center, y-coordinate	y_D	36
Disk radius	r	10

3-5. EXAMPLE 4. For the configuration of Figure 3-3 and Table 3-4, the ellipse-polygon overlap area is 501.859. The area of the polygon is 727.345. The area of the ellipse is 552.920.

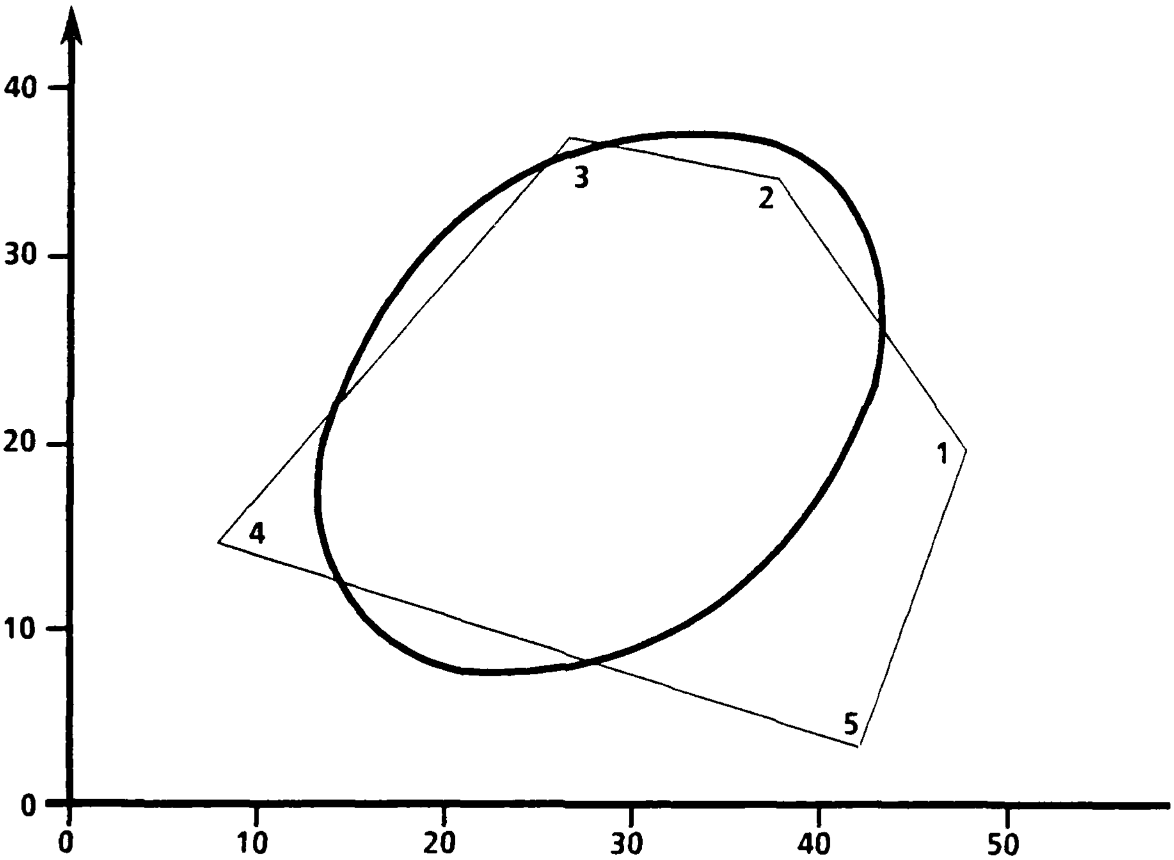

Figure 3-3. Configuration For Example 4

Table 3-4. Values for the Configuration of Example 4 (Figure 3-3)

Item	Symbol	Value
Ellipse center	XE	27.4
Ellipse center	YE	21.9
Ellipse semi-major axis	RX	16.0
Ellipse semi-minor axis	RY	11.0
Ellipse orientation angle	TDEG	43.5°
Polygon Vertex No. 1	PX(1)	47.0
Polygon Vertex No. 1	PY(1)	19.2
Polygon Vertex No.2	PX(2)	37.7
Polygon Vertex No. 2	PY(2)	32.5
Polygon Vertex No 3	PX(3)	26.2
Polygon Vertex No. 3	PY(3)	35.0
Polygon Vertex No. 4	PX(4)	7.9
Polygon Vertex No.4	PY(4)	14.5
Polygon Vertex No. 5	PX(5)	41.9
Polygon Vertex No. 5	PY(5)	3.0

5-6. EXAMPLE 5. Here we have to deal with a nonconvex polygon. There are two ways to do this.

a. The first way is to split the polygon into convex components and add the resultant areas. For the configuration of Example 5 (see Figure 3-4 and Table 3-5), this can be done by splitting the original polygon into two convex parts by a line joining vertices 3 and 7. The resulting areas will be as shown in Table 3-6.

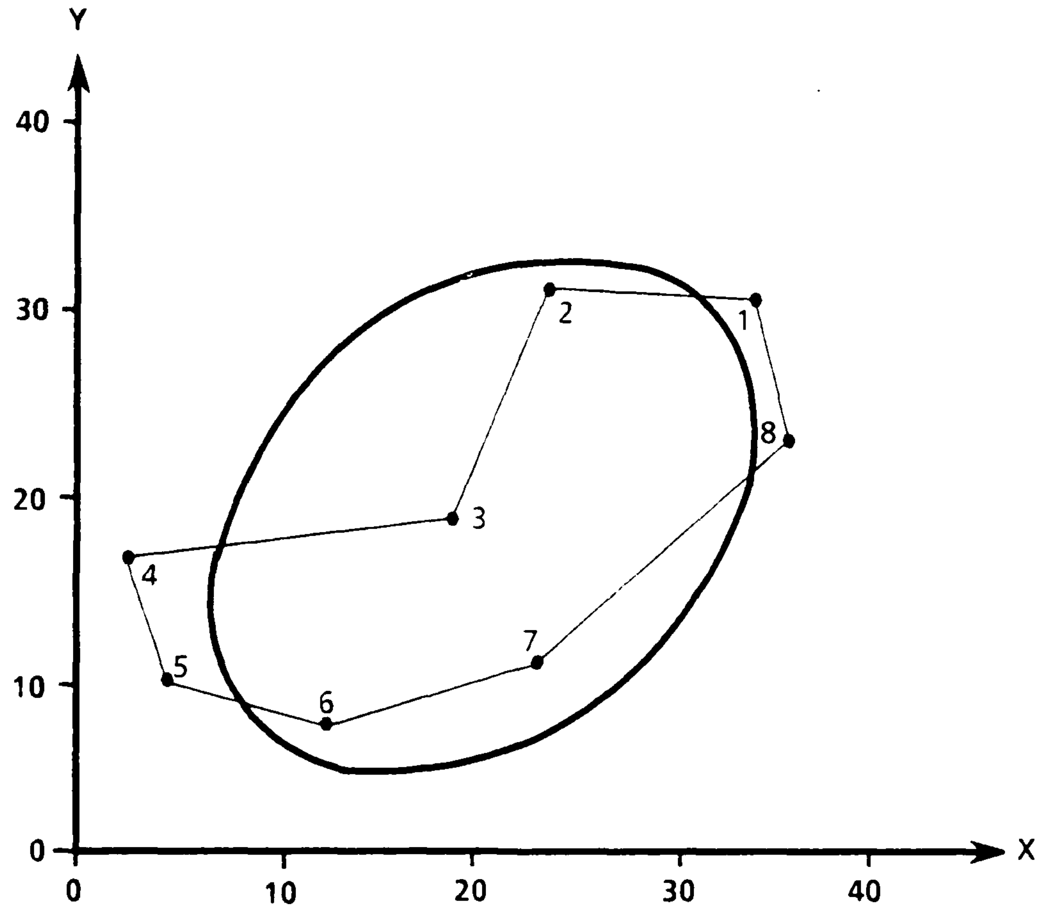

Figure 3-4. Configuration for Example 5

Table 3-5. Values for the Configuration of Example 5 (Figure 3-4)

Item	Symbol	Value
Ellipse center	XE	19.0
Ellipse center	YE	17.8
Ellipse semi-major axis	RX	16.0
Ellipse semi-minor axis	RY	11.0
Ellipse orientation angle	¯DEG	42.0°
Polygon Vertex No. 1	PX(1)	33.3
Polygon Vertex No. 1	PY (1)	29.8
Polygon Vertex No.2	PX(2)	23.9
Polygon Vertex No. 2	PY(2)	30.4
Polygon Vertex No 3	PX(3)	19.0
Polygon Vertex No. 3	PY(3)	19.0
Polygon Vertex No. 4	PX(4)	1.4
Polygon Vertex No.4	PY(4)	16.2
Polygon Vertex No. 5	PX(5)	3.8
Polygon Vertex No. 5	PY(5)	10.1
Polygon Vertex No. 6	PX (6)	11.5
Polygon Vertex No. 6	PY(6)	7.3
Polygon Vertex No. 7	PX(7)	23.8
Polygon Vertex No. 7	PY(7)	11.0
Polygon Vertex No. 8	PX(8)	35.5
Polygon Vertex No. 8	PY(8)	23.0

Table 3-6. Results of Method 1 for Example 5

Item	Area
Polygon 1-2-3-7-8	191.150
Polygon 3-4-5-6-7	170.665
Polygon 1-2-3-4-5-6-7-8	361.815
Ellipse overlap with polygon 1-2-3-7-8	170.231
Ellipse overlap with polygon 3-4-5-6-7	153.182
Ellipse overlap with polygon 1-2-3-4-5-6-7-8	323.413
Ellipse	552.920

b. The second way is to first take the convex hull of the vertices, and then subtract off areas that are within the convex hull but outside the original nonconvex polygon. The resulting areas will be as shown in Table 3-7.

Table 3-7. Results of Method 2 for Example 5

Item	Area
Polygon 1-2-4-5-6-7-8	445.275
Polygon 2-4-3	93.460
Polygon 1-2-3-4-5-6-7-8	361.815
Ellipse overlap with polygon 1-2-4-5-6-7-8	411.721
Ellipse overlap with polygon 2-4-3	88.308
Ellipse overlap with polygon 1-2-3-4-5-6-7-8	323.413
Ellipse	552.920

c. Either method gives the same final result. If a program is needed to generate general ellipse-polygon overlap areas, it is not clear which method would be easier to program.

CHAPTER 4

CONCLUSIONS AND OBSERVATIONS

4-1. RESULTS. A simple finite algorithm suffices to calculate the area of overlap of an ellipse and a (convex) polygon. The resultant algorithm is widely applicable in the sense that it gives theoretically exact answers for all possible configurations of the ellipse and the polygon. It is user-friendly in the sense that its implementation involves a straightforward computation whose net result is easily discernible. As a result, the algorithm should be easy to verify, debug, modify, or incorporate confidently into larger programs. Moreover, the accuracy of the calculated area of overlap can be assured by using double-precision arithmetic to control roundoff error.

4-2. OBSERVATIONS

a. Practical implementation of the algorithm would be aided by developing subroutines optimized for speed which could be incorporated into large simulations of wargames such as NUFAM, FORCEM, COSAGE, and others.

b. The method yields the exact area of overlap of an ellipse with a convex polygon. Since a triangle is convex, the method is, in principle, capable of being used to find the exact area of overlap of an ellipse with any region that can be triangulated, i.e., any polygon. Triangulation is not actually necessary--any decomposition of the polygonal region into a finite number of disjoint convex polygons is sufficient.

c. Many regions that arise in practice can be approximated by polygons. Hence the method can, in principle, be applied to approximate the area of overlap of an ellipse with a fairly arbitrary region of the sort that often arises in practice.

d. Although it may not be the most efficient algorithm for the purpose, the method can, in principle, be used to find the area of any polygon. All that is necessary is to find the area of overlap of the polygon with a disk whose radius is sufficiently large that it completely covers the polygon. If the area of overlap does not change when the disk's radius is increased slightly, then the radius is sufficiently large.

APPENDIX A

MATHEMATICAL DEVELOPMENT

A-1. INTRODUCTION

a. This appendix presents the mathematical developments on which the rest of the paper are based. Paragraph 1-7, Chapter 1, and Chapter 2 explained how the general ellipse-polygon overlap problem is reduced to that of finding the area of overlap between a disk and a wedge-shaped region. The development in this appendix shows how to find the vertex angle of the wedge-shaped region, how to locate the center of the disk with respect to the wedge, and how the overlap area is computed. Throughout this appendix, it is assumed that the polygon's vertices have been transformed to the values (X(J), Y(J)) by the affine transformation described in Chapter 2, and that the ellipse has been transformed to a disk of radius R centered at (0, 0).

b. Paragraphs A-5 and A-6 on the area of overlap of two disks and two rectangles are also included for reference, although no use of them is made elswhere in this paper.

A-2. FINDING THE WEDGE'S VERTEX ANGLE

a. Consider the wedge whose vertex coincides with the polygon's vertex J. By virtue of the orientation of the polygon (see paragraph 2-2a), vertex J has a preceding vertex JM (mnemonic for J-minus) and a succeeding vertex vertex JP (mnemonic for J-plus). The unit vector, E, from JM to J has components:

$$EX = (X(J) - X(JM)) / A$$

$$EY = (Y(J) - Y(JM)) / A$$

where A, the magnitude of the vector from JM to J, is given by:

$$A = SQR\ ((X(J) - X(JM))^2 + (Y(J) - Y(JM))^2).$$

The unit vector orthogonal to E and pointing generally toward the interior of the polygon has components

$$E' = (-EY, EX).$$

b. Think in terms of a local Cartesian coordinate system with origin at vertex J, x-axis along E, and y-axis along E'. Relative to this coordinate system, the coordinates (AX, AY) of vertex JP, found by taking the dot-products of the vector from vertex J to JP with E and with E', are:

AX = EX * (X(JP) - X(J)) + EY * (Y(JP) - Y(J))

AY = - EY * (X(JP) - X(J)) + EX * (Y(JP) - (Y(J))

c. Then the wedge angle GAMMA is found as:

GAMMA = ARG (AX, AY),

where ARG (U, V) is the angle from the x-axis to the line joining the origin to the point at (U, V). Note that the polygon is convex at vertex J if, and only if, GAMMA is less than PI radians.

A-3. FINDING THE COORDINATES OF THE DISK. The coordinates of the disk relative to the local coordinate system at vertex J are needed for subsequent developments. The local coordinates of a disk centered at coordinates (XD, YD) are found by taking the dot-products of the vector from vertex J to (XD, YD) with E and with E'. They are:

XO = EX * (XD - X(J)) + EY * (YD - Y(J))

YO = - EY * (XD - X(J)) + EX * (YD - Y(J)).

The polar coordinates (RHO, BETA) of the disk's center in the local coordinate system at vertex J are also useful. They are given by:

RHO = SQR ($XO^2 + YO^2$)

BETA = ARG (XO, YO).

Yet another local coordinate system is useful. Its origin is at vertex J. Its x-axis is directed in the positive direction from vertex J to vertex JP, and its y-axis is directed toward the interior of the wedge (and hence away from the interior of the polygon). The coordinates of the disk's center in this local coordinate system are:

XG = RHO * COS (GAMMA - BETA)

YG = RHO * SIN (GAMMA - BETA)

A-4. FINDING THE DISK-WEDGE OVERLAP AREA. THe disk-wedge overlap area is found by treating the different possible disk-wedge overlap cases that can arise. The possible cases of disk-wedge overlap (for the wedge at vertex J) are illustrated in Figure A-1, and are described in more detail below.

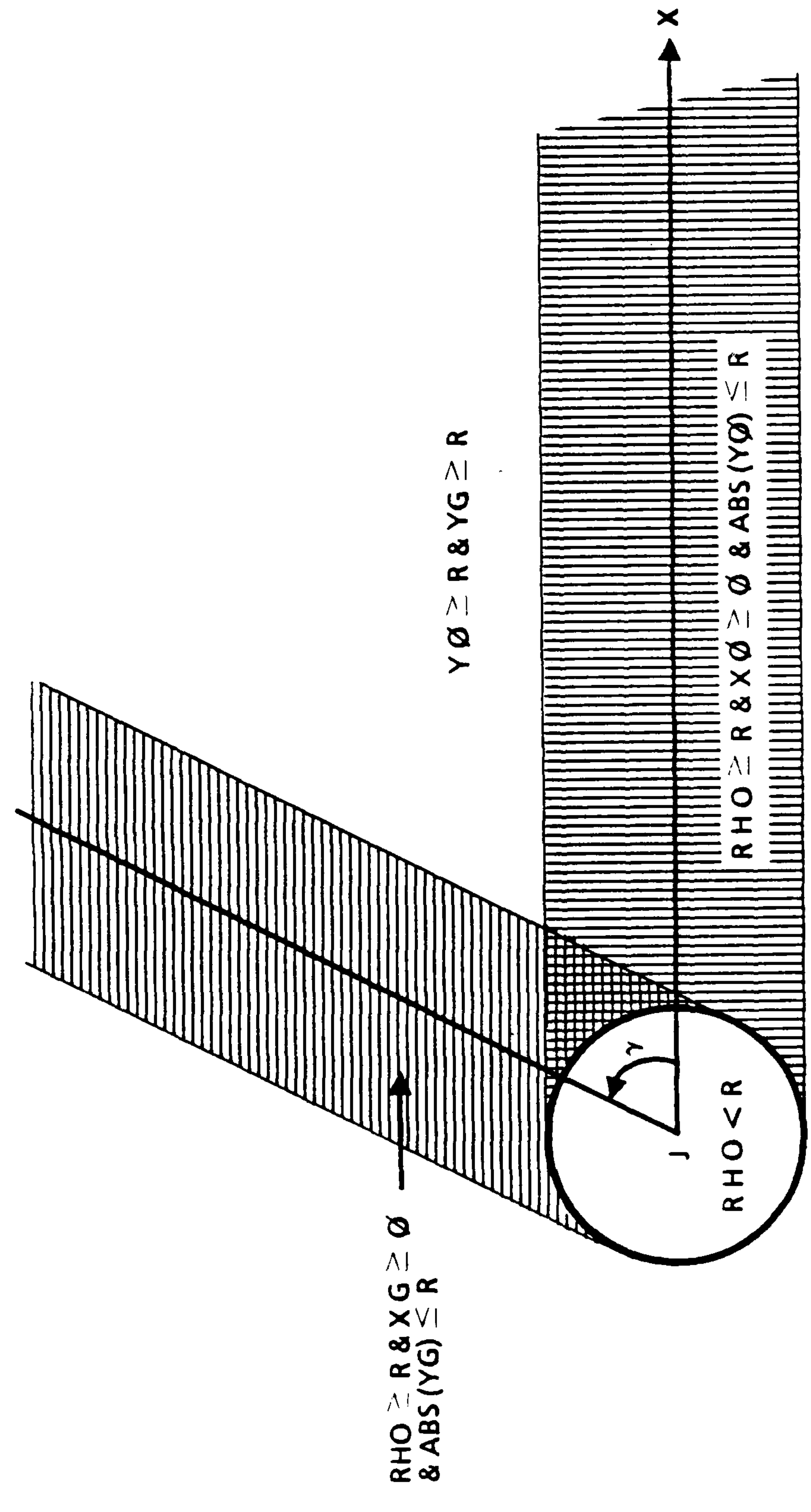

Figure A-1. Different Disk-Wedge Overlay Cases Possible

a. Case 1. Here YO ≥ R and YG ≥ R. Then the disk is completely contained within the wedge at vertex J, and the area of overlap is PI * R^2 .

b. Case 2. Here RHO ≥ R. And either XG ≥ 0 and ABS (YG) ≤ R, or XO ≥ R and ABS (YO) ≤ R, or both. Then the disk intercepts one or both edges of the wedge, but does not contain its vertex. Thus, there are either one or two segments of the disk that lie outside the wedge. The area of the disk-wedge overlap is equal to the area of the disk less the area of the segment (or segments) that lie outside the wedge. The area of the excluded segments are given by the algorithm:

AX = YO (or YG)

AY = SQR (R^2 - AX^2)

A = ARG (AX, AY)

ASEG = R^2 * (A - (SIN (2 * A)) / 2)

c. Case 3. Here the disk covers the vertex of the wedge, and RHO < R. This case is shown in Figure A-2. Here the center of the disk is located at (XO, YO). The rim of the disk intercepts the x-axis at coordinates (X1, 0), and intercepts the other edge of the wedge at coordinates (X2, Y2).

(1) We view the disk-wedge area of overlap as being composed of the triangle (0,0)-(X1, 0)-(X2, Y2) and the segment whose chord is (X1, 0)-(X2, Y2). The area of the triangle is

ATRIANGLE = X1 * Y2 /2.

The area of the segment is

ASEGMENT = R^2 * (ALF - (SIN (2 * ALF)) / 2),

where ALF is half the angle subtended by the chord at the center of the disk. It is equal to:

ALF = (T2 - T1) / 2,

where T2 and T1 are the angles shown in Figure A-2. If ALF is negative add PI radians to obtain a value between 0 and PI radians.

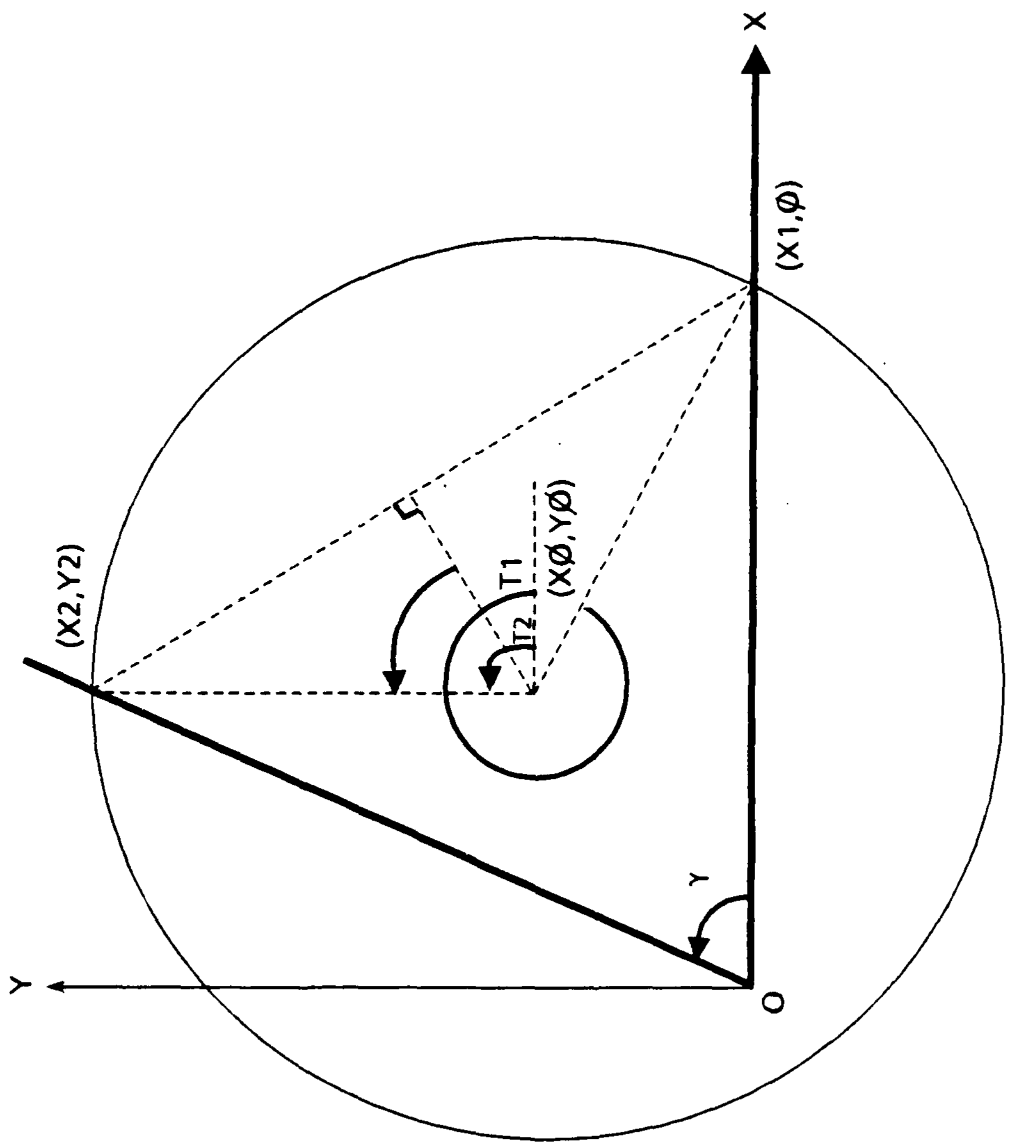

Figure A-2. Case 3-- Disk Covers the Vertex of Wedge

(2) We now explain how to find the angles T2 and T1, which completes the solution. We do this by first finding the coordinates of the points where the rim of the disk intercepts the X-axis and the other edge of the wedge. Solving the equation for the rim of the disk:

$$(X - XO)^2 + (Y - YO)^2 = R^2$$

for Y = 0, and letting X1 be the largest root, gives:

$$X1 = XO + SQR\ (R^2 - YO^2).$$

Then

$$T1 = ARG\ (X1 - XO,\ - YO).$$

Next, solving the equivalent equation for the rim of the disk:

$$(X - XG)^2 + (Y - YG)^2 = R^2$$

for YG = 0, and letting X be the largest root, gives:

$$X = XG + SQR\ (R^2 - YG^2),$$

and it follows immediately that

$$X2 = X * COS\ (GAMMA)$$

$$Y2 = X * SIN\ (GAMMA).$$

Then

$$T2 = ARG\ (X2 - XO,\ Y2 - YO).$$

d. Case 4. Here none of the preceding cases applies. Then the disk has zero area of overlap with the wedge for vertex J. This completes the solution for the disk-wedge overlap area.

A-5. AREA OF OVERLAP OF TWO RECTANGLES

a. Now suppose we want to find the area AF of overlap of two rectangles. We begin by letting

RROL (MX, MY, x, y, LX, LY)

be the area of overlap of two rectangles having the standard configuration shown in Figure A-3. Here both rectangles have sides parallel to the coordinate axes. The center of the first rectangle is at the origin. Its semi-length (dimension parallel to the x-axis) is MX and its semi-width (dimension parallel to the y-axis) is MY. The center of the second rectangle is at (x, y). Its semi-length is LX and its semi-width is LY.

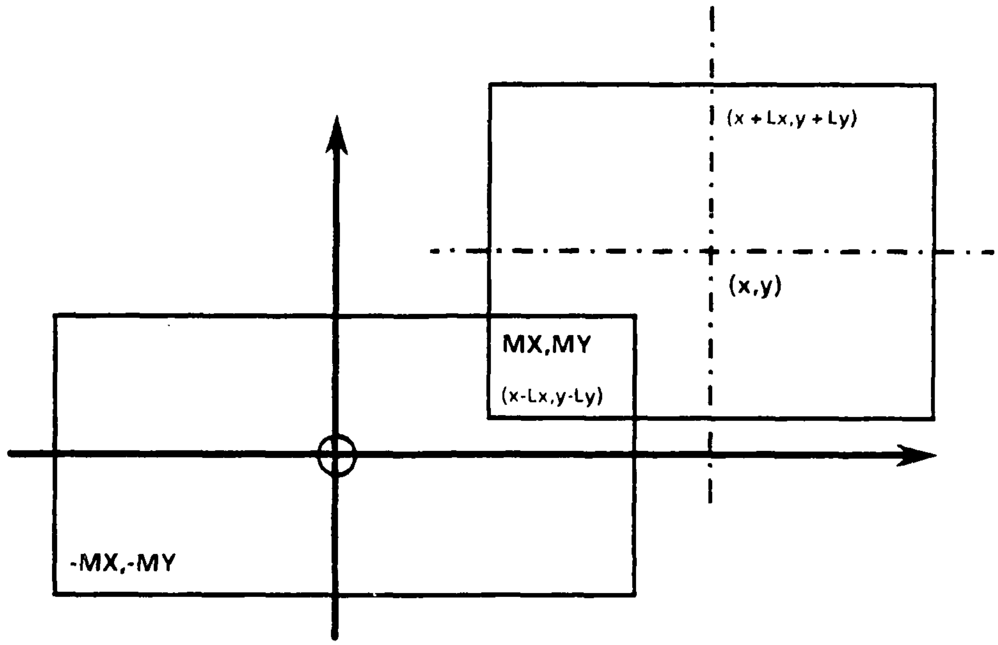

Figure A-3. Overlap of Two Rectangles (see text for explanation)

b. To find RROL, first let

SSOL (M, x, L)

be the length of overlap of two line segments having the standard configuration shown in Figure A-4. Here the first segment is centered at the origin and has semi-length M. The second segment is centered at x and has semi-length L. We first observe that

RROL(MX, MY, x, y, LX, LY) = SSOL(MX, x, LX) * SSOL(MY, y, LY),

and this reduces the computation of AF to finding a convenient algorithm for SSOL as a function of its arguments.

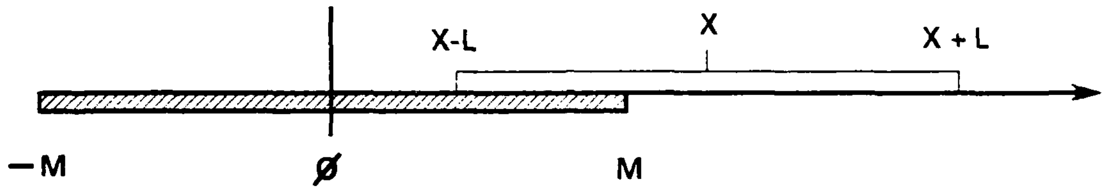

Figure A-4. Overlap of Two Line Segments

c. To find SSOL we proceed as follows.

$$SSOL\ (M,\ x,\ L) = SSOL\ (M,\ -x,\ L),$$

so we may, without loss of generality, replace x by ABS (x).

Also

$$SSOL\ (M,\ x,\ L) = SSOL\ (L,\ -x,\ M) = SSOL\ (L,\ x,\ M),$$

so we may, without loss of generality, assume that L is less than or equal to M. Finally, we observe that when L is less than or equal to M and x is nonnegative, it is easy to see that

$$SSOL = 2 * L, \text{ if } 0 \le x \le (M - L),$$
$$SSOL = [L + M - x]^+, \text{ if } (M - L) \le x,$$

where

$$[z]^+ = [z + ABS\ (z)]/2$$

is equal to z when z is nonnegative and zero otherwise. This algorithm gives SSOL as a function of its arguments and is used to evaluate RROL and from it the approximate area of overlap.

(1) The SSOL function can also be computed in the following manner. Let

V3 = ABS (M - L), and
V4 = M + L.

Then put

SSOL = 0, if V4 $\leq$ ABS (x)
SSOL = V4 - x, if V3 $\leq$ ABS (x) $\leq$ V4
SSOL = V4 - V3, if 0 $\leq$ ABS (x) $\leq$ V3.

This method is easily seen to be mathematically identical to the previous method, but may be easier to program.

(2) A simple program based on the second version might be as follows.

```
SUBROUTINE SSOL
ENTER WITH:  M, x, L
COMPUTE:

    V3 = ABS (M - L)
    V4 = M + L
    XABS = ABS (x)
    IF XABS ≥ V4 THEN SSOL = 0:  RETURN
    IF V4 > XABS > V3 THEN SSOL = V4 - XABS:  RETURN
    SSOL = V4 - V3:  RETURN
(END SUBROUTINE SSOL)
```

A-6. THE AREA OF OVERLAP OF TWO DISKS

a. Introduction. This paragraph presents formulas for the area of overlap of two disks. We start with the disks in general configuration as shown in Figure A-5. Here the larger disk has radius R and is centered at coordinates (x_R, y_R), while the smaller disk has radius r and is centered at (x_r, y_r). The separation between their centers is

$$d = \mathrm{SQR}\,[(x_R - x_r)^2 + (y_R - y_r)^2].$$

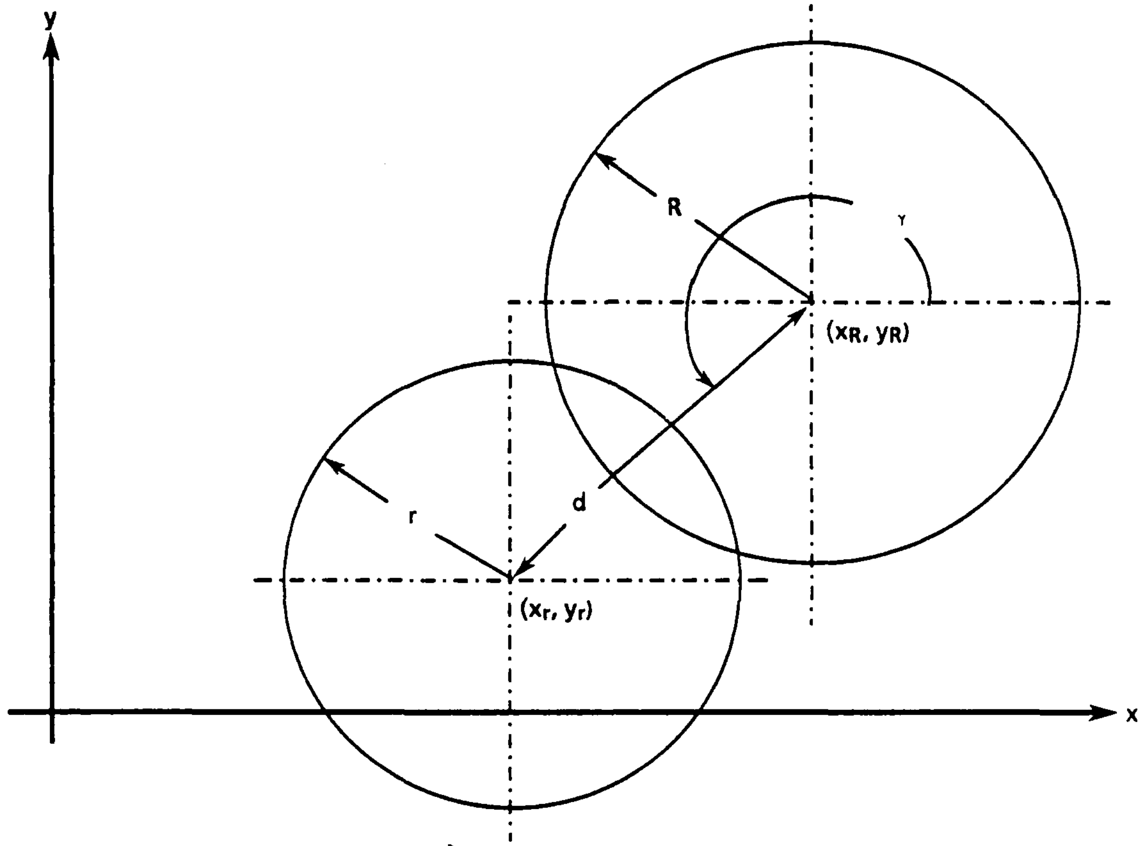

Figure A-5. Two Disks in General Configuration

The angle (gamma) is measured counterclockwise about the center of the larger disk, from the x-axis to the center of the smaller disk. It is equal to

$$(gamma) = ARG\ (x_r - x_R,\ y_r - y_R),$$

where ARG (u, v) is the angle from the x-axis to the point at coordinates (u, v). By convention, ARG (u,v) is greater than or equal to 0 and is less than (2 * pi) radians. If u = 0, then

$$ARG\ (u,v) = (pi\ /\ 2) * SGN(v)$$

radians. A rigid translation and rotation of coordinates, possibly followed by a reflection about the y-axis, will transform the general configuration to the standard configuration shown in Figure A-6.

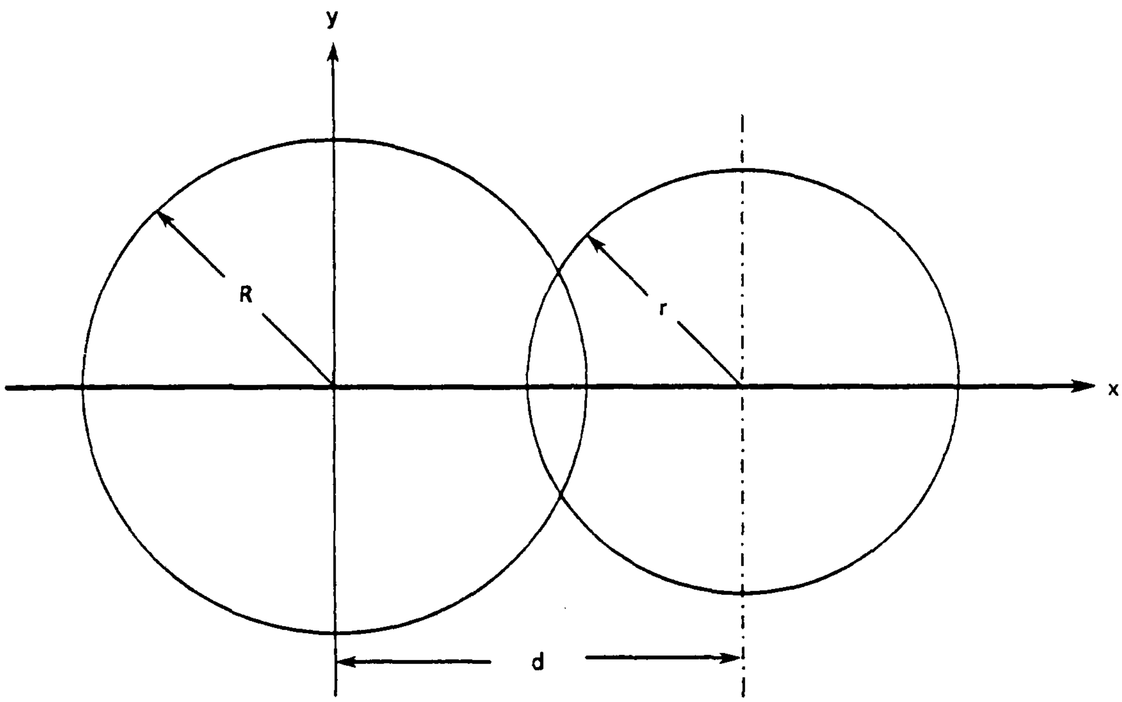

Figure A-6. Two Disks in Standard Configuration

In the standard configuration, the larger disk is centered at the origin and the smaller disk is centered at the point (d, 0). The transformation equations are

$$x_{C1} = (x_r - x_R) * \cos(\text{gamma}) + (y_r - y_R) * \sin(\text{gamma}),$$

and

$$d = ABS(x_{C1}) = SQR[(x_R - x_r)^2 + (y_R - y_r)^2].$$

The reduction to standard configuration clearly does not change the area of overlap. We now find formulas for the area of overlap of two disks in standard configuration.

b. Area of Overlap. We find the area of overlap by viewing it as the sum of the area AR bounded by BCDGB and the area Ar bounded by BFDGB, both of which are shown on Figure A-7.

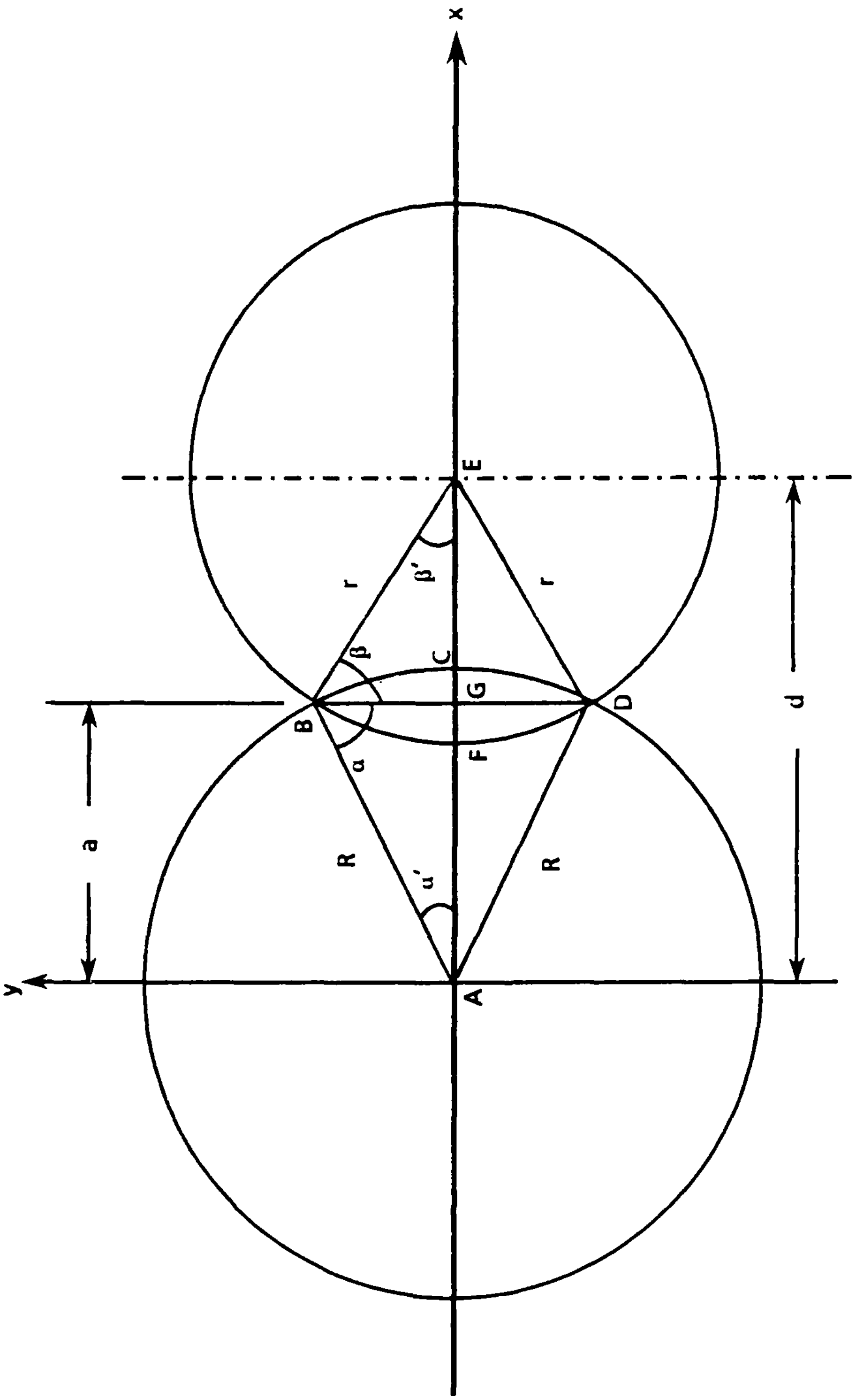

Figure A-7. Labeled Figure for the Overlap of Two Disks

Now the area BCDGB is equal to

$$AR = R^2 * [(alpha') - \sin(alpha') * \cos(alpha')],$$

as can easily be seen by noting that it is equal to the area of the sector ABCDA minus the area of the triangle ABDA. Similarly, the area bounded by BFDGB is

$$Ar = r^2 * [(beta') - \sin(beta') * \cos(beta')].$$

It is more convenient for further developments to express this in terms of the complementary angles (alpha) and (beta), where

$$(alpha) = (pi / 2) - (alpha')$$

$$(beta) = (pi / 2) - (beta').$$

Then we can write

$$AR = R^2 * [(pi / 2) - (alpha) - \sin(alpha) * \cos(alpha)]$$

and

$$Ar = r^2 * [(pi / 2) - (beta) - \sin(beta) * \cos(beta)].$$

The angles can be to and from

$$(alpha) = \mathrm{Arcsin}\,(a / R)$$

and

$$(beta) = \mathrm{Arcsin}\,[(d - a) / r],$$

where a is the length of the line segment AG. Here the Arcsin values are to be taken as less than or equal to (+pi/2) and greater than or equal to (-pi/2). The value of a can be determined by eliminating y between the simultaneous equations

$$a^2 + y^2 = R^2$$

and

$$(a - d)^2 + y^2 = r^2,$$

where y is the length of the line segment BG, to find

$$a = (R^2 - r^2 + d^2) / (2 * d).$$

c. **Algorithm.** Then the algorithm for DDOL, the area of overlap of two disks, is as follows. First find

$$d = SQR[(x_R - x_r)^2 + (y_R - y_r)^2],$$

and note that

$$DDOL = 0, \quad \text{when } R + r \leq d,$$

while

$$DDOL = (pi) * r^2, \quad \text{when } d \leq R-r.$$

The only remaining case is $R - r < d < R + r$. In that case, compute

$$a = (R^2 - r^2 + d^2) / (2 * d),$$

$$(alpha) = Arcsin\ (a / R), \text{ and}$$

$$(beta) = Arcsin\ [(d - a) / r].$$

Then find

$$AR = R^2 * [(pi / 2) - (alpha) - sin\ (alpha) * cos\ (alpha)],$$

$$Ar = r^2 * [(pi / 2) - (beta) - sin\ (beta) * cos\ (beta)],$$

and put

$$DDOL = AR + Ar.$$

APPENDIX B

COMPUTER PROGRAMS

B-1. INTRODUCTION. This appendix presents the listing and notes for a subroutine that implements the algorithms described in Appendix A and was used to obtain the results presented in Chapter 3.

B-2. PROGRAM NOTES

a. The subroutine is called EPOL, for "Ellipse-Polygon Overlap." Its listing is at paragraph B-3. Notes on it are provided below.

b. EPOL's inputs are as follows.

(1) XE = x-coordinate of the center of the ellipse.

(2) YE = y-coordinate of the center of the ellipse.

(3) RX = the semi-major axis of the ellipse.

(4) RY = the semi-minor axis of the ellipse.

(5) TDEG = orientation angle of the ellipse, in degrees. The orientation angle is the angle between the semi-major axis of the ellipse (prolonged if necessary) and the x-axis. It is conventionally restricted to be greater than -90 degrees and less than or equal to +90 degrees.

(6) (PX(J), PY(J)) = coordinates of the polygon's vertices, listed in counterclockwise order around the perimeter of the polygon. Here J runs from 1 to the number of vertices in the polygon.

c. EPOL's outputs are:

(1) DWOL(J) = disk-wedge overlap areas for each wedge. Here J runs from 1 to the number of vertices in the polygon. Note that these areas apply after the affine transformation described in Chapter 2 has reduced the ellipse to a disk.

(2) EPOL = ellipse-polygon overlap area.

d. EPOL is written in the APPLESOFT Basic language, which runs on APPLE II computers.

e. In line 3010, the call to subroutine 9030 prompts for the number N of vertices and for their coordinates PX(J), PY(J).

f. In line 3020, the call to subroutine 9110 prompts for the polygon's coordinates PX(J), PY(J).

g. In line 3110 and elsewhere, the call to subroutine 63960 returns the value of A, where A = ARG (AX, AY). Here AX and AY are values supplied to the subroutine and ARG (AX, AY) is the angle in radians from the x-axis to the point at coordinates (AX, AY). The value of A will be in the range from zero to 2* PI radians.

h. In line 3120 and elsewhere, the call to subroutine 63990 is a pause to allow the user to inspect the information displayed on the screen.

i. In line 3030 and elsewhere, PI = 3.14159265.

B-3. LISTING OF SUBROUTINE EPOL

```
3000  REM  -COMPUTE USING MNEMONICS
3010  IF N = 0 THEN GOSUB 9030: GOSUB3400:  GOTO 3030
3020  GOSUB 9110
3025  GOSUB 3400
3030  DPOL = PI * R^2:  FOR J = 1 TO N
3040  DWOL = 0
3050  JP = J + 1: IF JP > N THEN JP = 1
3060  JM = J - 1: IF JM < 1 THEN JM = N
3070  AX = X(J) - X(JM):AY = Y(J) -Y(JM):A = SQR (AX^2 + AY^2)
3080  EX = AX/A:EY = AY/A
3090  AX = EX * (X(JP) - X(J)) + EY * (Y(JP) - Y(J))
3100  AY = - EY * (X(JP) - X(J)) + EX * (Y(JP) - Y(J))
3110  GOSUB 63960:GAMMA = A
3120  IF GAMMA > PI THEN HOME : PRINT "POLYGON IS NOT CONVEX AT VERTEX
        ";J;"!": GOSUB 63990: RETURN
3130  AX = EX * (XD - X(J)) + EY * (YD - Y(J))
3140  AY = - EY * (SC - X(J)) + E X * (YD - Y(J))
3150  GOSUB 63960:BETA = A
3160  XO = AX:YO = AY:RHO = SQR  (AX^2 + AY^2)
3170  GOSUB 3500
3175  IF DWOL < 0 THEN DWOL = 0
3180  DPOL = DPOL - DWOL:DWOL(J) = DWOL
3185  IF DPOL < = 1E - 09 THEN D POL = 0: RETURN
3190  NEXT
3195  EPOL = DPOL/RAXIO
3199  RETURN
3400  REM  -REDUCE EPOL TO DPOL
3410  TRAD = PI * TDEG/180
3420  R = RY:RAXIO = RY/RX
3430  FOR J = 1 TO N
3440  X(J) = (PX(J) - XE) * COS (TRAD) + (PY(J) - YE) * SIN (TRAD)
3450  Y(J) = - (PX(J) - XE) * SIN (TRAD) + (PY(J) - YE) * COS (TRAD)
3460  X(J) = RAXIO * X(J)
3470  NEXT
3480  XD = 0 : YD = 0
3490  RETURN
3500  REM  -COMPUTE DWOL
3510  DWOL = 0:DA = PI * R^2
3520  IF YO < =  - R THEN RETURN
3540  IF TAN (GAMMA) = 0 THEN RETURN
3570  DEXTA = GAMMA - BETA:YG = RHO  *  SIN (DEXTA):XG = RHO * COS (DEXTA)
```

```
3590  IF YG < =  - R THEN RETURN
3590  IF YO = > R AND YG = > R THEN DWOL = DA:  RETURN
3600  IF RHO < R THEN GOSUB 3900 : RETURN
3610  IF XO < = 0 AND XG < = 0 THEN RETURN
3620  NIX = 0
3630  IF XO = > 0 AND ABS (YO) < = R THEN AX = YO:  GOSUB 370 0:NIX = NIX +
        ASEG
3640  IF XG = > 0 AND ABS (YG) < = R THEN AX = YG:  GOSUB 370 0:NIX = NIX +
       ASEG
3650  DWOL = DA - NIX: RETURN
3700  REM       ---COMPUTE AREA OF SEGMENT
3710  AY = SQR (R2 - AX 2):  GOSUB 63960
3720  ASEG = R2 * (A - SIN (2 * A)/2)
3730  RETURN
3800  REM    -VERTEX IN DISK CASE
3810 X1 = XO + SQR (R2 - YO2)
3820  AX = X1 - XO:AY = - YO:  GOSUB 63960:T1 = A
3830  A = XG + SQR (R2 - YG2) X2 = A * COS (GAMMA):Y2 = A * SIN (GAMMA)
3840  AX = X2 - XO:AY = Y2 - YO:  GOSUB
       63960:T2 = A
3850  ALF = (T2 - T1)/2
3860  IF ALF < 0 THEN ALF = ALF + PI: GOTO 3860
3870  IF ALF > PI THEN ALF = ALF - PI:  GOTO 3870
3880  DWOL = R2 * (ALF - SIN (2 * ALF)/2) + X1 * Y2/2
3890  RETURN
```

APPENDIX C

DISTRIBUTION

Addressee	No of copies
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZA Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZD Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZDF Washington, DC 20310	1
Deputy Chief of Staff for Personnel Headquarters, Department of the Army ATTN: DAPE-ZA Washington, DC 20310	1
Deputy Chief of Staff for Logistics Headquarters, Department of the Army ATTN: DALO-ZA Washington, DC 20310	1
Deputy Chief of Staff for Logistics Headquarters, Department of the Army ATTN: DALO-PLF Washington, DC 20310	1
Commander US Army Logistics Center Fort Lee, VA 23801	1

Addressee	No of copies
Deputy Under Secretary of the Army (Operations Research) Washington, DC 20310	2
Chief of Staff, Army ATTN: DACS-DMO Washington, DC 20310	1
Assistant Secretary of the Army (Research, Development, and Acquisition) Washington, DC 20310	2
Commander National Guard Bureau Room 2E394 The Pentagon Washington, DC 20310	1
Director US Army TRADOC Analysis Center White Sands Missile Range, NM 88002	2
Director US Army Materiel Systems Analysis Activity ATTN: AMXSY-LM (Mr. Fox) Aberdeen Proving Ground, MD 21005-5071	1
Director US Army Ballistic Research Laboratories Building 305 Aberdeen Proving Ground, MD 21005	2

Addressee	No of copies
Commander Combined Arms Combat Development Activity Fort Leavenworth, KS 66027	2
Commander Foreign Science and Technology Center 227th Street NE Charlottesville, VA 22901	1
Director Defense Nuclear Agency ATTN: LASS 6801 Telegraph Road Alexandria, VA 20305	2
Commander Army Research Institute 5001 Eisenhower Avenue Alexandria, VA 22333	1
Commander US Army Logistics Evaluation Agency New Cumberland Army Depot New Cumberland, PA 17070	1
Director Defense Logistics Studies Information Exchange US Army Logistics Management Center Fort Lee, VA 23801	1
Defense Technical Information Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314-6145	2

Addressee	No of copies
US Army Nuclear and Chemical Agency 7500 Backlick Road, Bldg #2073 ATTN: Library Springfield, VA 22150	2
Commander Air Mobility Research and Development Laboratory Eustis Directorate Fort Eustis, VA 23604	2
Commander US Army Mobility Equipment Research and Development Command Fort Belvoir, VA 22060	2
Commander US Army Research, Development, and Acquisition Information Systems Agency Radford, VA 24141	2
The Pentagon Library (Army Studies Section) ATTN: ANRAL-RS The Pentagon Washington, DC 20310	2
Commander US Army Forces Command ATTN: AFOP-OM Fort McPherson, GA 30330	1

Addressee	No of copies
Director Program Analysis and Evaluation Office of the Secretary of Defense Room 2E330 The Pentagon Washington, DC 20310	2
Organization of the Joint Chiefs of Staff ATTN: J-8 Room 1D936, The Pentagon Washington, DC 20310	2
Joint Deployment Agency Deploying Systems Division MacDill Air Force Base, FL 33608	2
Commandant US Army War College Carlisle Barracks, PA 17013	2
Commandant US Army War College ATTN: Library Carlisle Barracks, PA 17013	1
Air University ATTN: AU CADRE/WGTA (Capt. Taylor) Maxwell Air Force Base, AL 36112-5000	2
Commandant US Air War College Maxwell Air Force Base, AL 36112-5532	2

Addressee	No of copies
Commandant US Navy War College Newport , RI 02840	2
President National Defense University ATTN: NDU-LD-CDC Washington, DC 20319-6000	2
Commandant Armed Forces Staff College Norfolk, VA 23511	2
Commandant US Army Command and General Staff College Fort Leavenworth, KS 66027	2
Commandant US Army Command and General Staff College ATTN: Department of Combat Development, Force Development Fort Leavenworth, KS 66027	2
Superintendent United States Military Academy ATTN: Mathematics Department West Point, NY 10996	2
Superintendent United States Military Academy ATTN: Engineering Department West Point, NY 10996	2
Superintendent Naval Postgraduate School Monterey, CA 93940	2

Addressee	No of copies
Naval Postgraduate School ATTN: Department of Operations Research Code 55PY (Professor Parry) Monterey, CA 93940	1
Commandant US Army Infantry School ATTN: ATSH-IVT Fort Benning, GA 31905	1
Commandant US Army Armor School ATTN: ATSB-CD Fort Knox, KY 40121-5215	1
Commandant US Army Field Artillery School Fort Sill, OK 73503	1
Commandant US Army Air Defense School Fort Bliss, TX 79916	1
Commandant US Army Aviation School Fort Rucker, AL 36360	1
Commandant US Army Engineer School Fort Belvoir, VA 22060	1
Commandant US Army Transportation School Fort Eustis, VA 23604	1

Addressee	No of copies
Commandant US Army Intelligence Center and School ATTN: ATSI-TD Fort Huachuca, AZ 85613	1
Commander in Chief United States Readiness Command ATTN: RCDA MacDill Air Force Base, FL 33608	1
Commander US Army Combat Developments Experimentation Command Fort Ord, CA 93941	1
Director US Army Human Engineering Laboratory Aberdeen Proving Ground, MD 21005-5001	1
CINCPAC Staff C3S ASII Box 32A Camp Smith, HI 96861	1
Commander US Army Western Command ATTN: APOP-SPM Fort Shafter, HI 96858-5100	1
Commander US Army Information Systems Command Fort Belvoir, VA 22060	1

Addressee	No of copies
Commander Eighth US Army APO San Francisco 96301	1
Commander US Army, Japan ATTN: AJCS APO San Francisco 96343	1
Deputy Chief of Staff for Intelligence Headquarters, Department of the Army Washington, DC 20310	1
Commander, USAITAC AIAIT-HI, Tech Info Bldg 203, STOP 314 Washington Navy Yard Washington, DC 20374	1
Commander/Director US Army Engineer Studies Center Casey Building, No. 2594 Fort Belvoir, VA 22060	1
Commander US Army Corps of Engineers 20 Massachusetts Avenue, NW Washington, DC 20314-1000	1
Commander US Army Missile Command Redstone Arsenal, AL 35898-5090	1

Addressee	No of copies
Commander US Army Ballistic Missile Defense Systems Command Huntsville, AL 35807	1
Commander in Chief US Army, Europe & Seventh Army ATTN: AEAGF APO New York 09403	1
Commander in Chief US Army, Europe & Seventh Army ATTN: AEAGX-OR (Dr. Leake) APO New York 09403	1
Commander US Army Training and Doctrine Command ATTN: ATCD-AU Fort Monroe, VA 23651	1
Commander US Army Training and Doctrine Command Fort Monroe, VA 23651	1
Commander US Army Materiel Command 5001 Eisenhower Avenue Alexandria, VA 22333	1
Commander US Army Tank-Automotive Command Warren, MI 48090	1
Commander US Army Information Systems Command Fort Huachuca, AZ 85613	1

Addressee	No of copies
US Army CE Command Program Analysis and Evaluation Systems Analysis Division Fort Monmouth, NJ 07703	1
Air Force Center for Studies and Analyses AFCSA/SAMI Room 1D363, Pentagon Washington, DC 20330-5425	2
Headquarters, US Air Force Office of Worldwide Management of Studies & Analyses ATTN: AF/SAL The Pentagon Washington, DC 20330	1
Air University Command Readiness Exercise System (CRES) ATTN: Program Management Office Maxwell Air Force Base, AL 36112	1
Commander Tactical Air Command TAC Analysis Group Langley Air Force Base, VA 23665	1
Joint Studies Group (TAC) ATTN: OSS Langley Air Force Base, VA 23665	1
Commander USAF Systems Command Andrews Air Force Base Washington, DC 20334	1

Addressee	No of copies
Commandant Air Force Institute of Technology ATTN: AFIT-EN Wright-Patterson Air Force Base, OH 45433	1
Headquarters Rome Air Development Center (AFSC) ATTN: COAM (Mr. T. Humiston) Griffiss Air Force Base, NY 13441	1
Center for Wargaming Naval War College ATTN: Naval Warfare Gaming System Code 33 Newport, RI 02840	1
President Center for Naval Analyses 4401 Ford Avenue Post Office Box 16268 Alexandria, VA 22302-0268	2
Naval Research Laboratory ATTN: Code 5704 4555 Overlook Avenue Washington, DC 20375	2
Chief of Naval Operations ATTN: OP-96 Room 4A526, Pentagon Washington, DC 20350	2
Commander Naval Air Systems Command ATTN: Mission and Effectiveness Analysis Division (AIR-5264) Washington, DC 20361	1

Addressee	No of copies
Commander Military Sealift Command Ship Operations Branch 4228 Wisconsin Avenue Washington, DC 20390	1
Commander Naval Sea Systems Command Code 09B5 Washington, DC 20361	1
US Army Liaison Officer Naval Weapons Center China Lake, CA 93555	1
US LNO Supreme Allied Commander Atlantic Norfolk, VA 23511-5100	1
Marine Corps Operations Analysis Group Center for Naval Analyses 4401 Ford Avenue P. O. Box 11280 Alexandria, VA 22302-0268	2
Director USMC Development and Education Center Quantico, VA 22134	2

Internal Distribution:

Helmbold	15
DeRiggi	15
Unclassified Library	2

GLOSSARY

1. MODELS, ROUTINES, AND SIMULATIONS

CEM	Concepts Evaluation Model
COSAGE	Combat Sample Generator
FORCEM	Force Evaluation Model
NUFAM	Nuclear Fire Planning and Assessment Model
VIC	VECTOR in COMMANDER

2. DEFINITIONS

$[z]^+$	An expression whose value is z if z is greater than zero, and zero otherwise.
(relational expression)	An expression whose value is plus one if the relational expression is true, and zero otherwise. For example, $(2 < 3) = +1$, and $(9 < 5) = 0$.
ABS (z)	Absolute value of z.
ARG (u, v)	The angle about the origin to the point at coordinates (u, v). If u is equal to zero, then ARG (u, v) = SGN (v)* (pi/2). By convention, ARG (u, v) is greater than or equal to zero, and is less than (2 * pi) radians.
MIN (x, y)	The lesser of x and y.
ROOF (z)	The least integer greater than or equal to z.
SGN (a)	The signum function of a, i.e., +1 if a is greater than zero, 0 if a is equal to zero, and -1 if a is less than zero.
SQR (z)	Square root of z.

AN ALGORITHM FOR CALCULATING THE AREA OF OVERLAP OF AN ELLIPSE AND A CONVEX POLYGON	STUDY SUMMARY CAA-RP-87-4

THE REASON FOR PERFORMING THIS STUDY was to develop and document an improved algorithm for determining in computer simulations the area of overlap of an ellipse and a convex polygon.

THE PRINCIPAL FINDINGS are that a useful algorithm can be developed for determining in computer simulations the area of overlap of an ellipse and a convex polygon. It appears to be a new method offering many advantages over those previously proposed.

THE MAIN ASSUMPTION is that the polygon is convex.

THE PRINCIPAL LIMITATION is that numerical roundoff error may, under some conditions, reduce the accuracy of the result.

THE SCOPE OF THE WORK is limited to determining the area of overlap of an ellipse and a convex polygon.

THE RESEARCH OBJECTIVE was to develop and document an algorithm for determining in computer simulations the area of overlap of an ellipse and a convex polygon.

THE WORK WAS SUPPORTED BY the US Army Concepts Analysis Agency.